Anforderungsbasierte Erkennung von Feature-Interaktionen in der Produktlinienentwicklung

Andreas Metzger, Klaus Pohl

Software Systems Engineering, Institut für Informatik & Wirtschaftsinformatik (ICB)
Universität Duisburg-Essen, Schützenbahn 70, 45117 Essen
{metzger, pohl}@sse.uni-due.de

Abstract: Feature-Interaktionen sind unerwünschte Wechselwirkungen zwischen Produktmerkmalen. Wegen der typischerweise sehr hohen Zahl an Applikationen, die aus den wiederverwendbaren Artefakten einer Produktlinie abgeleitet werden können, ist es nicht möglich, alle potenziellen Applikationen einzeln auf Feature-Interaktionen hin zu analysieren. Der hier vorgestellte RAFINA-Ansatz zur anforderungsbasierten Erkennung von Feature-Interaktionen nutzt die Eigenschaften von Produktlinienvariabilität und von speziellen Feature-Interaktionen, um eine sehr kleine Teilmenge repräsentativer Applikationen auszuwählen, die einen Rückschluss auf die Feature-Interaktionen der gesamten Produktlinie erlauben.

1. Motivation

Eine Feature-Interaktion ist eine Situation, in welcher das Verhalten eines Systems insgesamt nicht die Spezifikation seiner *einzelnen* Features (Produktmerkmale) erfüllt [Gi97], die Features isoliert aber korrekt gegenüber ihrer jeweiligen Spezifikation sind. Verursacht werden diese speziellen Fehler durch *unerwünschte* Wechselwirkungen zwischen den einzelnen Produktmerkmalen [KK98]. Wegen der kombinatorischen Explosion der Anzahl möglicher Abläufe in einem System mit sehr vielen Features bieten einfache Testverfahren und manuelle Inspektionen keine Lösung für das systematische Aufdecken von Feature-Interaktionen [CKM03]. Beim Testen können nicht alle möglichen Fälle erschöpfend überprüft werden [Li02, S. 35] und viele Feature-Interaktionen zeigen sich oft erst bei komplexeren Wechselwirkungen oder in Situationen, die nicht unbedingt offensichtlich sind. Auch Inspektionen und Reviews sind nicht sonderlich gut geeignet, um Fehler in dynamischen Wechselwirkungen aufzudecken [Li02, S. 287f.]. Wegen der Defizite dieser „traditionellen" Qualitätssicherungsansätze treten viele Feature-Interaktionen häufig erst während des Betriebs zu Tage.

Dieses Problem wird in der Produktlinienentwicklung (vgl. z.B. [PBL05]) deutlich verstärkt, da nicht nur ein einzelnes System entwickelt wird, sondern durch Wiederverwendung generischer Artefakte eine hohe Zahl an Applikationen erstellt wird. Die späte Erkennung von Feature-Interaktionen kann zu sehr hohen Kosten führen, wenn zu dem Zeitpunkt des Auftretens einer Feature-Interaktion bereits mehrere Applikationen der Produktlinie entwickelt wurden und diese dann möglicherweise alle korrigiert werden müssen. Der in dieser Arbeit vorgestellte RAFINA-Ansatz zur anforderungsbasierten Erkennung von Feature-Interaktionen in Produktlinien ermöglicht es, Feature-Interaktionen bereits während der Entwicklung der wiederverwendbaren Artefakte zu

erkennen. Hierzu werden die Spezifika der Produktlinienvariabilität und die Eigenschaften einer speziellen Klasse von Feature-Interaktionen ausgenutzt.

In Abschnitt 2 wird zunächst das Komplexitätsproblem bei der Feature-Interaktionserkennung in Produktlinien erläutert. Anschließend wird in Abschnitt 3 der RAFINA-Ansatz vorgestellt und in Abschnitt 4 die Komplexitätsreduktion des Ansatzes diskutiert.

2. Feature-Interaktionen in Produktlinien

Bei der Produktlinienentwicklung (vgl. z.B. [PBL05]) werden im Domain Engineering zunächst wiederverwendbare Artefakte erstellt, welche dann im Application Engineering zur Ableitung individueller Applikationen genutzt werden. Die Variabilität der Produktlinie wird in den wiederverwendbaren Artefakten realisiert, welche durch die sog. Bindung der Variabilität an die produktspezifischen Bedürfnisse angepasst werden. Um bereits im Domain Engineering Feature-Interaktionen erkennen zu können, muss die Produktlinienvariabilität bei der Interaktionserkennung berücksichtigt werden.

Die beiden zentralen Modellierungskonzepte, welche eine explizite Dokumentation von Variabilität einer Produktlinie erlauben, sind *Variationspunkt* und *Variante* [PBL05]. Ein Variationspunkt definiert, an welcher Stelle sich Applikationen einer Produktlinie unterscheiden können. Die konkreten Ausprägungen sind die jeweils zu einem Variationspunkt assoziierten Varianten. Diese Varianten entsprechen dabei typischerweise den Features, welche die Applikationen der Produktlinie aufweisen können.

Varianten können zu einer *alternativen Auswahl* zusammengefasst werden, wobei die zulässige Anzahl an wählbaren Varianten durch die Angabe einer Multiplizität (analog zur UML) ausgedrückt werden kann. Für eine alternative Auswahl aus einer Menge von n möglichen Varianten (also variablen Features) mit einer Multiplizität von $j..k$ berechnet sich die Anzahl der möglichen Kombinationen zu:

$$K_{\text{alle}}(j,k,n) := \sum_{i=j}^{k} \binom{n}{i}$$

Viele Produktlinien besitzen eine sehr hohe Anzahl an Variationspunkten, die z.B. im Automobilbereich weit über 100 liegen kann. Angenommen, jeder dieser Variationspunkte erlaubt jeweils die alternative Auswahl (1..*) aus drei möglichen Varianten (bzw. Features). Wegen $K_{\text{alle}}(1, 3, 3) = 7$ Möglichkeiten pro Variationspunkt („*“ wird mit n gleichgesetzt) sind dies bereits $7^{100} \approx 10^{85}$ mögliche Kombinationen, die jeweils ein konkretes Produkt repräsentieren. Selbst wenn nur ein Bruchteil dieser Kombinationen sinnvolle Produkte darstellt, wäre eine „Brute-Force“-Strategie, welche alle potenziellen Applikationen einzeln auf Feature-Interaktionen hin analysiert, nicht anwendbar.

3. Anforderungsbasierte Erkennung von Feature-Interaktionen

Die Lösungsidee für den hier vorgestellten RAFINA-Ansatz (Requirements-bAsed Feature INteraction Analysis) basiert auf einer geschickten Auswahl von repräsentativen Applikationen (*Repräsentanten*) einer Produktlinie, wodurch eine drastische Reduktion der Komplexität des Erkennungsansatzes erreicht wird. Zunächst werden die Repräsen-

tanten durch die Selektion geeigneter Varianten bestimmt (siehe Abschnitt 3.1). Ausgehend von diesen Varianten werden dann Anforderungsmodelle aus den wiederverwendbaren Anforderungsartefakten durch Bindung der Variabilität abgeleitet. Die so erhaltenen Modelle entsprechen denen aus der Einzelsystementwicklung und können mit existierenden Ansätzen auf Feature-Interaktionen hin untersucht werden (siehe z.B. [CKM03], [Me04]).

3.1 Repräsentantenauswahl

Die Auswahl geeigneter Repräsentanten stellt den zentralen Beitrag unseres Ansatzes dar. Dies gelingt durch Nutzung der Spezifika der Produktlinienvariabilität, insbesondere der Multiplizitäten der alternativen Auswahl (vgl. Abschnitt 2) und der Eigenschaften sog. *m*-Wege Feature-Interaktionen, einer speziellen Art von Feature-Interaktionen.

Eine *m*-Wege Feature-Interaktion tritt nicht zwischen weniger als *m* Features sondern nur zwischen genau *m* Features auf. So impliziert z.B. eine 3-Wege Feature-Interaktion zwischen den Features A, B und C, dass zwischen A & B, zwischen B & C und zwischen A & C keine 2-Wege Feature-Interaktionen bestehen. In der Praxis wurden bisher einige wenige *m*-Wege Feature-Interaktionen in der Telekommunikation beobachtet (siehe [KO03]). Im Bereich der Eingebetteten Systeme konnten wir in verschiedenen Fallstudien bislang keine solchen Feature-Interaktionen entdecken (vgl. [Me04]).

Für die Auswahl der Repräsentanten nehmen wir nun an, dass es in dem betrachteten System keine *m*-Wege Interaktionen mit $m > 2$ gibt. Dies impliziert, dass wenn in einem Repräsentanten eine Interaktion zwischen *r* Features aufgedeckt wird, paarweise zwischen allen diesen Features 2-Wege Feature-Interaktionen bestehen. Um eine maximale Reduktion der Anzahl der zu betrachtenden Repräsentanten zu erreichen, wählen wir daher diejenigen Repräsentanten aus, die eine jeweils maximal mögliche Anzahl an gebundenen Varianten aufweisen. Denn werden Feature-Interaktionen in einem solchen „maximalen“ Repräsentanten erkannt, dann finden sich entsprechende Feature-Interaktionen auch in allen „kleineren“ Applikationen, welche nur eine Teilmenge der Varianten des Repräsentanten beinhalten. Hätte man z.B. in einem Repräsentanten eine Feature-Interaktion zwischen X, Y und Z festgestellt, dann würden Interaktionen auch in Applikationen auftreten, in denen z.B. nur X und Y oder nur Y und Z gebunden sind.

Die Auswahl von Repräsentanten im RAFINA-Ansatz erfolgt daher durch die Selektion der jeweils maximalen Zahl (k) an zulässigen Varianten pro alternativer Auswahl. Als Folge reduziert sich die Anzahl der Kombinationsmöglichkeiten pro alternativer Auswahl auf $K_{rep}(k, n) = n$ über k. Für das KFZ-Beispiel ergibt sich eine drastische Reduktion von 7^{100} auf 1, da man für jeden Variationspunkt jeweils die maximale Anzahl an Varianten wählen kann, also jeweils ein $K_{rep}(3,3) = 1$ erhält. Die Auswirkung einer Verletzung der obigen Annahme werden wir in Abschnitt 4 diskutieren.

3.2 Beispiel

In Abb. 1 ist ein Variabilitätsmodell für eine Gebäudeautomationsproduktlinie als OVM-Diagramm dargestellt (Orthogonale Variabilitäts-Modellierung, siehe [PBL05]). Ein

Variationspunkt wird hierbei durch ein Dreieck, eine *Variante* durch ein Rechteck und eine *alternative Auswahl* durch einen Kreisbogen dargestellt.

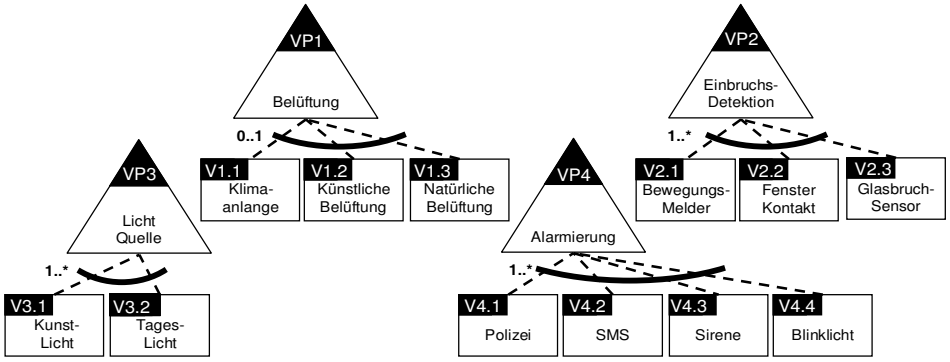

Abb. 1: Variabilitätsmodell einer Gebäudeautomationsproduktlinie

Diese Produktlinie erlaubt die Definition von $8_{VP1} \cdot 7_{VP2} \cdot 3_{VP3} \cdot 15_{VP4} = 2520$ Produkten. Mit dem RAFINA-Ansatz müssen von diesen 2520 möglichen Applikationen lediglich $3_{VP1} \cdot 1_{VP2} \cdot 1_{VP3} \cdot 1_{VP4} = 3$ Repräsentanten analysiert werden.

4. Diskussion des Ansatzes

Die deutliche Komplexitätsreduktion des RAFINA-Ansatzes wird durch die Auswahl der jeweils maximal möglichen Anzahl an Varianten pro alternativer Auswahl erreicht. Je näher diese maximale Anzahl (k) dabei an der tatsächlichen Anzahl Varianten (n) liegt, umso gravierender wird die Komplexitätsreduktion gegenüber einem „Brute-Force"-Ansatz. Je weiter die maximale Anzahl von Varianten bei 1 liegt, desto weniger aufwändig ist auch der „Brute-Force"-Ansatz und der Gewinn durch unseren optimierten RAFINA-Ansatz fällt geringer aus. In Abb. 2 sind konkrete Werte für die Anzahl der Kombinationen für ein $n = 13$ gezeigt.

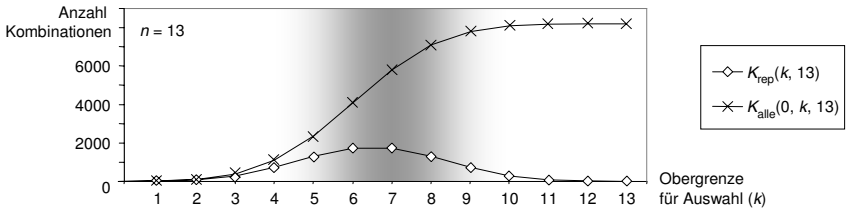

Abb. 2: Anzahl der zu untersuchenden Kombinationen

Besitzt eine Produktlinie viele Variationspunkte, bei denen die maximale Zahl an wählbaren Varianten zu einem mittleren Bereich tendiert (grau schattierter Bereich in Abb. 2), dann kann auch in unserem Ansatz die Anzahl der zu untersuchenden Applikationen sehr groß werden. Daher kann es angebracht sein, die maximale Zahl wählbarer Varianten (also k) für die Analyse wie im Folgenden beschrieben künstlich zu verändern.

Eine Veränderung in Richtung 1 ist hierbei nur bis $k = 2$ sinnvoll, da ansonsten Interaktionen nicht erkannt werden, welche durch eine unerwünschte Wechselwirkung zwischen den einzelnen Varianten einer alternativen Auswahl bestehen. Ideal wäre eine Vergrößerung von k auf n, da damit die deutlichste Komplexitätsreduktion, auf $K_{rep}(n, n) = 1$, zu erreichen ist. Hierbei ist jedoch zu beachten, dass zusätzliche Interaktionen, die in der „echten" Produktlinie nicht auftreten, fälschlicherweise erkannt werden könnten. Diese lassen sich jedoch automatisch eliminieren, indem man für die jeweiligen unerwünscht interagierenden Features (d.h. Varianten) feststellt, ob diese in einer gültigen Applikation der Produktlinie überhaupt gemeinsam gebunden werden können. Durch die „künstlichen" Vergrößerung von k entsteht jedoch ein weitaus schwierigeres Problem. Eine Bindung einer nicht vorgesehenen Auswahl an Varianten (z.B. sich ausschließende Alternativen) kann zu ungültigen Anforderungsmodellen führen, welche dann nicht als Eingabe für einen Feature-Interaktionserkennungsalgorithmus geeignet sind.

Handelt es sich bei einer erkannten Feature-Interaktion entgegen der in Abschnitt 3.1 getroffenen Annahme um eine m-Wege Feature-Interaktion, dann bedeutet dies, dass in Applikationen, welche nicht alle diese m Varianten beinhalten, keine Feature-Interaktion auftreten würden. Das heißt man bekommt lediglich ein falsch-positives Ergebnis geliefert (Fehler 1. Art). Der RAFINA-Ansatz garantiert daher in jedem Fall, dass alle echten Feature-Interaktionen einer Produktlinie erkannt werden (abhängig von dem eingesetzten Erkennungsalgorithmus aus der Einzelsystementwicklung).

Gibt man den Anspruch an eine garantierte Aufdeckung aller Feature-Interaktionen auf, dann besteht eine weitere Möglichkeit, die Komplexität in den Griff zu bekommen. Mit der Annahme, dass m-Wege Feature-Interaktionen (mit $m > 2$) nicht existieren bzw. nicht erkannt werden müssen, ist es nur notwendig, alle 2-Wege Feature-Interaktionen aufzudecken. Dazu genügt es jeweils alle zweier Kombinationen von Varianten zu überprüfen, was bei einer Anzahl von m Varianten insgesamt zu $m \cdot (m-1)$ Kombinationen führt. Im KFZ-Beispiel sind das dennoch $100 \cdot 99 = 9900$ Kombinationen, d.h. der ursprüngliche RAFINA-Ansatz mit nur einem Repräsentanten ist hier immer noch deutlich besser. Allerdings kann der weitergehende Ansatz Sinn machen, wenn wir uns im mittleren, kritischen, Bereich von k befinden.

5. Verwandte Arbeiten

Es existieren bislang nur wenige Beiträge, die sich dem Problem der Feature-Interaktionen in Software-Produktlinien widmen. Robak et al. [RF01], Riebisch et al. [RSP03], de Bruin et al. [BV01] und Gurp et al. [GBS04] heben das Problem zwar hervor, diskutieren aber keine detaillierten Lösungen zur Erkennung von Feature-Interaktionen.

Interaktionen zwischen Features einer Produktlinie lassen sich durch entsprechende Abhängigkeiten zwischen den Features spezifizieren (siehe z.B. [GBS04]). Lee et al. führen in [LK04] sog. operationale Abhängigkeiten ein, diese sind z.B. „Verwendung" und „Modifikation". Ferber et al. beschreiben von der "Umgebung induzierte Interaktionen" [FHS02]. Alle diese Beiträge bieten allerdings keine konkreten Techniken zur Erkennung von Feature-Interaktionen bzw. der Abhängigkeiten zwischen den Features.

6. Zusammenfassung und Ausblick

Feature-Interaktionen sind eine spezielle Klasse von Fehlern. Mit dem in diesem Beitrag vorgestellten RAFINA-Ansatz können Feature-Interaktionen bereits sehr früh in der Produktlinienentwicklung erkannt und damit frühzeitig aufgelöst werden. Die Komplexität der Erkennung wird durch RAFINA drastisch ggü. einem „Brute-Force"-Ansatz reduziert. Die Erkennung ist vollständig automatisierbar und Erkennungsalgorithmen aus der Einzelsystementwicklung können je nach Domäne (z.B. Eingebettete Systeme), Zweck und Art der Anforderungsmodelle austauschbar eingesetzt werden.

Nicht alle Kombinationen von möglichen Varianten liefern zulässige Produkte, z.B. macht ein Cabrio mit dem „Feature" Schiebedach keinen Sinn. Gegenstand zukünftiger Forschung ist daher, solche Einschränkungen bei der Erkennung zu berücksichtigen.

Literaturverzeichnis

[BV01] Bruin, H. de; Vliet, H. van: Scenario-Based Generation and Evaluation of Software Architectures. In (Bosch, J. Hrsg.) Proceedings GCSE 2001. Springer, Heidelberg, 2001; S. 128–139.

[CKM03] Calder, M.; Kolberg, M.; Magill, E.; et al.: Feature Interaction: A Critical Review and Considered Forecast. Computer Networks, 41, 2003; S. 115–141.

[FHS02] Ferber, S.; Haag, J.; Savolainen, J.: Feature Interaction and Dependencies: Modeling Features for Reengineering a Legacy Product Line. In (Chastek, G. Hrsg.) Proceedings SPLC 2002. Springer, Heidelberg, 2002; S. 235–256.

[GBS04] Gurp, J.v.; Bosch, J.; Svahnberg, M. Managing Variability in Software Product Lines. In Proceedings Working WICSA 2001, IEEE Computer Society, 2001; S. 45–54.

[Gi97] Gibson, J.P.: Feature Requirements Models: Understanding Interactions. In (Dini, P.; Boutaba, R.; Logrippo, L.) Proceedings FIW IV, IOS Press, Amsterdam, 1997.

[KK98] Keck, D.O.; Kuehn, P.J.: The Feature and Service Interaction Problem in Telecommunications Systems: A Survey. IEEE Transactions on Software Engineering. 24(10), 1998; S. 779–796.

[KO03] Kawauchi, S.; Ohta, T. Mechanism for 3-way Feature Interactions Occurrence and a Detection System Based on the Mechanism. In (Amyot, D.; Logrippo, L. Hrsg.) Proceedings FIW VII. IOS Press, Amsterdam, 2003; S. 313–327.

[LK04] Lee, K.; Kang, K.C.: Feature Dependency Analysis for Product Line Component Design. In (Bosch, J.; Krueger, C. Hrsg.) Proceedings ICSR 2004, Springer, Heidelberg, 2004; S. 69–85.

[Li02] Liggesmeyer, P.: Software-Qualität.Spektrum Akademischer Verlag, Heidelberg, 2002

[Me04] Metzger, A.: Feature Interactions in Embedded Control Systems. Computer Networks, 45(5), 2004; S. 625–644.

[PBL05] Pohl, K.; Böckle, G.; van der Linden, F.: Software Product Line Engineering – Foundations, Principles, and Techniques. Springer, Heidelberg, 2005.

[RF01] Robak, S.; Franczyk, B.: Feature Interaction and Composition Problems in Software Product Lines. In Proceedings ECOOP 2001 Workshop #08. Budapest, June, 2001.

[RSP03] Riebisch, M.; Streitferdt, D.; Pashov, I.: Modeling Variability for Object-Oriented Product Lines. In (Buchmann, A.; Buschmann, F. Hrsg.) ECOOP 2003 Workshop Reader. Springer, Heidelberg, 2003.